1. What is NoSQL data base?

A NoSQL (originally referring to "non SQL", "non relational" or "not only SQL"). It provides a mechanism for storage and retrieval of data which is modeled in means other than the tabular relations used in relational databases.

NoSQL is an approach to databases that represents a shift away from traditional relational database management systems (RDBMS) that rely on tables, columns, rows, or schemas to organize and retrieve data. NoSQL databases on the other hand do not rely on these structures and use more flexible data models. As RDBMS have increasingly failed to meet the performance, scalability, and flexibility needs that next-generation, data-intensive applications require, NoSQL databases have been adopted by mainstream enterprises. NoSQL is particularly useful for storing unstructured data, which is growing far more rapidly than structured data and does not fit the relational schemas of RDBMS.

Features of NoSQL:

* Generic data model
  + Heterogeneous containers, including sets, maps, and arrays
* Dynamic type discovery and conversion
  + NoSQL analytics systems support runtime type identification and conversion so that custom business logic can be used to dictate analytic treatment of variation.
* Non-relational and De-normalised
  + Data is stored in single tables as compared to joining multiple tables.
* Commodity hardware
  + Adding more of the economical servers allows NoSQL databases to scale to handle more data.
* Highly distributable
  + Distributed databases can store and process a set of information on more than one device.

2. How does data get stored in NoSQL database?

There are different ways to store data in NoSQL database.

Document Oriented Databases (MongoDB): Stores documents as a data and allows indexing of documents on the basis of not only its primary identifier but also its properties

Graph Based Databases (Neo4j): A graph database uses graph structures with nodes, edges, and properties to represent and store data.

Column Based Databases (HBase): The column-oriented storage allows data to be stored effectively. It avoids consuming space when storing nulls by simply not storing a column when a value doesn’t exist for that column.

Key Value Databases (Membase): The key of a key/value pair is a unique value in the set and can be easily looked up to access the data. Key-value NoSQL databases which is very useful for an application to support high-speed read and write processing of non-transactional data. Stored values can be any type of binary object (text, video, JSON document, etc.) and are accessed via a key. The application has complete control over what is stored in the value, making this the most flexible NoSQL model. Data is partitioned and replicated across a cluster to get scalability and availability. For this reason, key value stores often do not support transactions. However, they are highly effective at scaling applications that deals with high-velocity, non-transactional data.

3. What is a column family in HBase?

Collection of columns having multiple actual columns inside. In the HBase data model columns are grouped into column families, which must be defined up front during table creation. Column families are stored together on disk, which is why HBase is referred to as a column-oriented data store.

4. How many maximum number of columns can be added to HBase table?

There is no hard limit to number of columns in HBase , we can have more than 1 million columns but usually three column families are recommended ( not more than three). HBase currently does not do well with anything above two or three column families so it is recommended to keep the number of column families in the schema low.

5. Why columns are not defined at the time of table creation in HBase?

To maintain the dynamic behavior of columns, columns are not defined at the time of table creation. For example if we are having address as a column family, we can have different column qualifiers such as city, state etc. at run time. Column families are defined rather than defining columns since columns are dynamic in HBase, which can be different for each records.

6. How does data get managed in HBase?

Data in HBase is stored in Tables and these Tables are stored in Regions. When a Table becomes too big, the Table is partitioned into multiple Regions. These Regions are assigned to Region Servers across the cluster. Each Region Server hosts roughly the same number of Regions.

* Region servers serve data for reads and writes.
* HBase Master process handles the Region assignment, DDL (create, delete tables) operations
* Zookeeper maintains a live cluster state.

7. What happens internally when new data gets inserted into HBase table?

• HBase Tables are divided horizontally by row key range into “Regions.”

• A region contains all rows in the table between the region’s start key and end key.

• Regions are assigned to the nodes in the cluster, called “Region Servers,” and these serve data for reads and writes.

• A region server can serve about 1,000 regions.

Each Region Server contains a Write-Ahead Log (called HLog) and multiple Regions. Each Region in turn is made up of a MemStore and multiple StoreFiles (HFile). The data lives in these StoreFiles in the form of Column Families. The MemStore holds in-memory modifications to the Store (data).